

ANSI/CEA Standard

Digital Television (DTV) Closed Captioning

ANSI/CEA-708-E

August 2013



CEA
Consumer Electronics Association®

Documents
www.CE.org
Document Sharing Hub

NOTICE

Consumer Electronics Association (CEA®) Standards, Bulletins and other technical publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards, Bulletins and other technical publications shall not in any respect preclude any member or nonmember of CEA from manufacturing or selling products not conforming to such Standards, Bulletins or other technical publications, nor shall the existence of such Standards, Bulletins and other technical publications preclude their voluntary use by those other than CEA members, whether the standard is to be used either domestically or internationally.

Standards, Bulletins and other technical publications are adopted by CEA in accordance with the American National Standards Institute (ANSI) patent policy. By such action, CEA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard, Bulletin or other technical publication.

This document does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this document to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

This document is copyrighted by the Consumer Electronics Association (CEA®) and may not be reproduced, in whole or part, without written permission. Federal copyright law prohibits unauthorized reproduction of this document by any means. Organizations may obtain permission to reproduce a limited number of copies by entering into a license agreement. Requests to reproduce text, data, charts, figures or other material should be made to CEA.

(Formulated under the cognizance of the CEA **R4.3 Television Data Systems Subcommittee.**)

Published by
©CONSUMER ELECTRONICS ASSOCIATION 2013
Technology & Standards Department
www.CE.org

All rights reserved

CONTENTS

1 Scope.....	1
1.1 Overview	1
1.2 Notation	1
1.2.1 Binary Notation	2
1.2.2 Hex Notation	2
1.2.3 Equals and Assignment	2
1.2.4 Bitstream Syntax Notation	2
1.2.5 Acronyms & Mnemonics	2
1.2.6 Terms and Definitions	3
2 Normative References	3
2.1 Normative Reference List	4
2.1.1 Normative Reference Acquisition	4
2.2 Informative References	4
2.2.1 Informative Reference List	4
2.2.2 Informative Reference Acquisition	5
2.3 Compliance Notation	5
3 Caption Channel Layered Protocol	6
4 DTVCC Transport Layer	9
4.1 DTVCC Transport Channel Data Rate	9
4.2 Pre-Allocated Bandwidth	9
4.3 DTV cc_data() Syntax and Semantics	9
4.3.1 Structure and Interpretation of cc_data()	11
4.3.2 CEA-608 Datastream	12
4.3.3 DTVCC Caption Channel Packets in cc_data()	14
4.3.4 Caption Order in Progressive and Interlaced Frames	15
4.3.5 Padding Unused Space in cc_data()	15
4.3.6 Frame Rates	15
4.3.7 Example cc_data() Contents	17
4.4 Latency	18
4.4.1 Zero Delay Progressive Sequences	19
4.4.2 Zero Delay Interlaced Sequences	19
4.4.3 End-to-end Delay	19
4.4.4 Encoding Delay	19
4.4.5 Decoding Delay	19
4.4.6 Intermediate Processing Delay	19
4.5 Caption Service Metadata	19
4.5.1 Decoder Processing of Metadata	20
4.6 Shortened Caption Channel Packets	21
4.7 Independent Control of Audio and Caption Services	21
5 DTVCC Packet Layer	21
5.1 Caption Channel Packet Syntax and Semantics	21
5.2 CCP Processing	22
6 DTVCC Service Layer	23
6.1 Services	23
6.2 Service Blocks	24
6.2.1 Standard Service Block Header	24
6.2.2 Extended Service Block Header	25

6.2.3 Null Service Block Header	25
6.2.4 Service Block Data	25
6.2.5 Service Blocks within Caption Channel Packets	25
6.3 Transport Constraints on Encapsulating Caption Data	26
7 DTVCC Coding Layer - Caption Data Services (Services 1 - 63).....	27
7.1 Code Space Organization	27
7.1.1 Extending the Code Space	29
7.1.2 Unused Codes.....	30
7.1.3 Numerical Organization of Codes	30
7.1.4 Code Set C0 - Miscellaneous Control Codes.....	30
7.1.5 C1 Code Set - Captioning Command Control Codes	32
7.1.6 G0 Code Set - ASCII Printable Characters	33
7.1.7 G1 Code Set - ISO 8859-1 Latin-1 Character Set	34
7.1.8 G2 Code Set - Extended Miscellaneous Characters	35
7.1.9 G3 Code Set - Future Expansion.....	36
7.1.10 C2 Code Set - Extended Control Code Set 1.....	37
7.1.11 C3 Code Set - Extended Control Code Set 2.....	38
8 DTVCC Interpretation Layer	42
8.1 DTVCC Caption Components.....	42
8.2 Screen Coordinates.....	42
8.3 User Options	44
8.4 Caption Windows.....	44
8.4.1 Window Identifier	45
8.4.2 Window Priority.....	45
8.4.3 Anchor Points	45
8.4.4 Anchor ID.....	45
8.4.5 Anchor Location	46
8.4.6 Window Size	46
8.4.7 Window Row and Column Locking.....	47
8.4.8 Word Wrapping	48
8.4.9 Window Text Painting	49
8.4.10 Window Display	51
8.4.11 Window Colors and Borders	51
8.4.12 Predefined Window and Pen Styles.....	52
8.5 Caption Pen.....	52
8.5.1 Pen Size	52
8.5.2 Pen Spacing	53
8.5.3 Font Styles.....	53
8.5.4 Character Offsetting	54
8.5.5 Pen Styles.....	54
8.5.6 Foreground Color and Opacity.....	54
8.5.7 Background Color and Opacity.....	54
8.5.8 Character Edges	54
8.5.9 Caption Text Function Tags	56
8.5.10 Pen Attributes	57
8.6 Caption Text.....	57
8.7 Caption Positioning.....	58
8.7.1 Location within Internal Buffer.....	58
8.7.2 Location (0,0).....	58
8.7.3 Caption Row Lengths.....	58
8.8 Color Representation	58
8.9 Service Synchronization	58
8.9.1 Delay Command.....	59
8.9.2 DelayCancel Command.....	59

8.9.3 Reset Command.....	59
8.9.4 Reset and DelayCancel Command Recognition.....	60
8.9.5 Service Reset Conditions	61
8.10 DTVCC Command Set	61
8.10.1 Window Commands	62
8.10.2 Pen Commands.....	63
8.10.3 Synchronization Commands	63
8.10.4 Caption Text	63
8.10.5 Command Descriptions	63
8.11 Proper Order of Data	84
8.11.1 Simple Roll-up Style Captions.....	84
8.11.2 Simple Paint-on Style Captions.....	84
8.11.3 Simple Pop-on Style Captions.....	85
9 DTVCC Decoder Manufacturer Requirements and Recommendations	85
9.1 DTVCC Section 6.1 - Services	85
9.2 DTVCC Section 6.2 - Service Blocks	85
9.2.1 Caption Service Directory and DTVCC Services	85
9.2.2 Decoding 16 Services.....	86
9.2.3 Selecting CEA-608 Services Regardless of Presence of Caption Service Directory	86
9.2.4 Ignoring Reserved Field in caption_service_descriptor()	86
9.2.5 Automatic Switching from 708 to 608.....	86
9.3 DTVCC Section 7.1 - Code Space Organization	86
9.4 DTVCC Section 8.2 - Screen Coordinates	87
9.5 DTVCC Section 8.4 - Caption Windows.....	89
9.6 DTVCC Section 8.4.2 - Window Priority.....	89
9.7 DTVCC Section 8.4.6 - Window Size	89
9.8 DTVCC Section 8.4.8 - Word Wrapping	89
9.9 DTVCC Section 8.4.9 - Window Text Painting	89
9.9.1 Justification.....	89
9.9.2 Print Direction	90
9.9.3 Scroll Direction	90
9.9.4 Scroll Rate	90
9.9.5 Smooth Scrolling	90
9.9.6 Display Effects	90
9.10 DTVCC Section 8.4.11 - Window Colors and Borders	91
9.11 DTVCC Section 8.4.12 - Predefined Window and Pen Styles.....	91
9.12 DTVCC Section 8.5.1 - Pen Size	91
9.13 DTVCC Section 8.5.3 - Font Styles.....	91
9.14 DTVCC Section 8.5.4 - Character Offsetting	91
9.15 DTVCC Section 8.5.5 - Pen Styles.....	91
9.16 DTVCC Section 8.5.6 - Foreground Color and Opacity.....	91
9.17 DTVCC Section 8.5.7 - Background Color and Opacity.....	91
9.18 DTVCC Section 8.5.8 - Character Edges	91
9.19 DTVCC Section 8.8 - Color Representation	91
9.20 Character Rendition Considerations	92
9.21 DTVCC Section 8.9 - Service Synchronization	93
9.22 DTV to NTSC (CEA-608) Transcoders	93
9.23 Receivers Without Displays and Set-top Box (STB) Options	94
9.24 Use of CEA-608 datastream by DTV Receivers	94
10 DTVCC Authoring and Encoding for Transmission (Informative)	94
10.1 Caption Authoring and Encoding	95
10.2 Monitoring Captions.....	96
Annex A Possible Decoder Implementations (Informative).....	97

Annex B Transmission	98
B.1 Interpretation of Transmission Syntax	98
Annex C Caption Channel Packet Transmission Examples in MPEG-2 Video (Informative)	99
C.1 PICTURE 1: picture_structure = 11, top_field_first = 1, repeat_first_field = 1	99
C.2 PICTURE 2: picture_structure = 11, top_field_first = 0, repeat_first_field = 0	99
C.3 PICTURE 3: picture_structure = 11, top_field_first = 0, repeat_first_field = 1	100
Annex D Transmission Order and Display Process Examples in MPEG-2 Video (Informative)	101
Annex E DTVCC in the ATSC Transport with MPEG-2 Video (Informative)	102
E.1 General	102
E.2 MPEG-2 Picture User Data	103
E.2.1 Latency	103
E.3 Caption Service Metadata and PSIP	103
E.4 Caption Service Encoding	103
Annex F (Deleted)	104
Annex G Closed Caption Data Structure	105

Figures

Figure 1 DTV Closed-Captioning Protocol Model	8
Figure 2 cc_data() State Table	12
Figure 3 Example of CEA-608 Captioning Field Buffers	13
Figure 4 Caption Channel Packet	21
Figure 5 CCP State Table	23
Figure 6 Service Block	24
Figure 7 Service Block Header	24
Figure 8 Extended Service Block Header	25
Figure 9 Null Service Block Header	25
Figure 10 Service Blocks in a Caption Channel Packets (Example)	26
Figure 11 Example of Window and Grid Location	43
Figure 12 DTV 16:9 Screen and DTVCC Window Positioning Grid	44
Figure 13 Anchor ID Location	45
Figure 14 Implied Caption Text Expansion Based on Anchor Points	46
Figure 15 Examples of Caption Window Shrinking when User Selects Small Character Size	47
Figure 16 Examples of Caption Window Growing when Going to Larger Font	48
Figure 17 Examples of Various Justifications, Print Directions and Scroll Directions	50
Figure 18 Character Background Color Examples	54
Figure 19 Edge Type Examples	56
Figure 20 Reset & DelayCancel Command Detector(s) and Service Input Buffers	60
Figure 21 Reset & DelayCancel Command Detector(s) Detail	61
Figure 22 Minimum Grid Location Super Cell Example	88
Figure 23 Caption Authoring and Encoding into Caption Channel Packets	95
Figure 24 Relationship Between Caption Data and Frames	96
Figure 25 DTVCC Transport Stream Decoder for an MPEG-2 Transport	97
Figure 26 DTVCC Caption Data in the DTV Bitstream	102
Figure 27 Structure of cc_data()	105

Tables

Table 1 DTVCC Protocol Stack.....	6
Table 2 cc_data() Syntax.....	10
Table 3 Closed-Caption Type (cc_type) Coding.....	11
Table 4 DTVCC Example #1 - MPEG-2 Video Transport Channel—cc_data() parameters.....	16
Table 5 DTVCC Example #2 - MPEG-2 Video Transport Channel—cc_data() parameters.....	17
Table 6 Aligned cc_data() structure and CCP Example.....	17
Table 7 Unaligned Caption Channel Packet Example.....	18
Table 8 cc_data() Structure Example Showing Unusual Sequences of cc_valid	18
Table 9 DTVCC Caption Channel Packet Syntax.....	22
Table 10 Service Block Syntax.....	24
Table 11 DTVCC Code Space Organization	28
Table 12 DTVCC Code Set Mapping	29
Table 13 C0 Code Set	30
Table 14 C1 Code Set	32
Table 15 G0 Code Set.....	33
Table 16 G1 Code Set.....	34
Table 17 G2 Code Set.....	35
Table 18 G3 Code Set.....	36
Table 19 C2 Code Set	37
Table 20 Extended Codes and Bytes to Skip—C2 Code Set.....	38
Table 21 C3 Code Set	38
Table 22 Extended Codes & Bytes to Skip—C3 Code Set.....	39
Table 23 Extended Codes and Bytes to Skip 0x90-0x9F	41
Table 24 Cursor Movement After Drawing Characters	50
Table 25 Safe Title Area and Recommended Character Dimensions	53
Table 26 Predefined Window Style IDs.....	68
Table 27 Predefined Pen Style IDs.....	69
Table 28 G2 Character Substitution Table	87
Table 29 Screen Coordinate Resolutions & Limits	87
Table 30 Minimum Color List Table	91
Table 31 Alternative Minimum Color List Table	92
Table 32 Caption Channel Packet Transmission Example A.....	99
Table 33 DTVCC Caption Channel Packet Transmission Example B	99
Table 34 DTVCC Caption Channel Packet Transmission Example C	100

(This page intentionally left blank.)

FOREWORD

This standard defines a method for coding text with associated parameters to control its display. This document specifies the standard for Closed Captioning in Digital Television (DTV) technology. Predecessors of this document were developed under the auspices of the Consumer Electronics Association (CEA) Technology & Standards R4.3 Television Data Systems Subcommittee in parallel with the U.S. Advanced Television Systems Committee's (ATSC) definition, design, and development of the audio, video and ancillary data processing standard for Advanced Television. The DTV standard developed by the cable industry in SCTE for caption carriage is documented in SCTE 21 [6].

CEA-708-E supersedes CEA-708-D.

(This page intentionally left blank.)

Digital Television (DTV) Closed Captioning

1 Scope

This standard defines DTV Closed Captioning (DTVCC) and provides specifications and guidelines for caption service providers, distributors of television signals, decoder and encoder manufacturers, DTV receiver manufacturers, and DTV signal processing equipment manufacturers. CEA-708-E may also be useful in other systems. This standard includes the following:

- a) a description of the transport method of DTVCC data in the DTV signal
- b) a specification for processing DTVCC information
- c) a list of minimum implementation recommendations for DTVCC receiver manufacturers
- d) a set of recommended practices for DTV encoder and decoder manufacturers

The use of the term DTV throughout is intended to include, and apply to, High Definition Television (HDTV) and Standard Definition Television (SDTV).

1.1 Overview

DTVCC is a migration of the closed-captioning concepts and capabilities developed in the 1970's for National Television Systems Committee II (NTSC) television video signals to the digital television environment defined by the ATV (Advanced Television) Grand Alliance and standardized by ATSC. This new television environment provides for larger screens and higher screen resolutions, as well as higher data rates for transmission of closed-captioning data.

NTSC Closed Captioning (CC) consists of an analog waveform inserted on line 21, field 1 and possibly field 2, of the NTSC Vertical Blanking Interval (VBI). That waveform provides a transport channel which can deliver 2 bytes of data on every field of video. This translates to a nominal 60 or 120 bytes per second (Bps), or a nominal 480 or 960 bits per second (bps).

In contrast, DTV Closed Captioning is transported as a logical data channel in the DTV digital bitstream. DTV-specific closed captioning is allocated 9600 bps for each program in some applications. This increased capacity opens the possibility for simultaneous transmission of captions in multiple languages and with multiple reading levels, as well as the transport of an entire CEA-608 [1] datastream¹.

The DTV standard also accommodates a variety of increased horizontal and vertical resolutions (e.g., 704x480, 1280x720 and 1920x1080), versus the single 525 vertical scan line format for NTSC. These added resolutions provide for more defined representations of character fonts and other on-screen objects.

The heart of any DTVCC caption display is the caption "window," which is similar to the *window* concept found in many computer Graphical User Interfaces (GUIs). Windows are placed within the DTV screen, and caption text is placed within windows. Windows and text have a variety of color, size and other attributes.

This standard describes the above issues in a reverse-hierarchical (i.e., low-to-high level) fashion. It follows an "Open Systems Interconnect (OSI) Reference Model"-type protocol stack for layered protocols. DTVCC consists of 5 protocol layers: the Transport Layer, the Packet Layer, the Service Layer, the Coding Layer, and the Interpretation Layer. The discussion of the first 2 layers is a detailed presentation of data organization issues. The discussion of the last 2 layers provides a more informative presentation of the unique aspects of closed captioning. Some readers may wish to start with these last 2 layers first, beginning in Section 7.

1.2 Notation

Designers should interpret this standard's syntax and values based on notational conventions taken from the referenced Motion Picture Experts Group (MPEG), ATSC and CEA standards. Numbering and

¹ CEA-608 datastream is a generic term used to mean all valid datastreams from before the original EIA-608 of 1993 on through the current CEA-608 [1].

counting loops generally begin from zero unless otherwise specified. Some specific cases and examples are provided in the following sections to assist in clarifying notations used herein.

1.2.1 Binary Notation

The following notational convention is used in this standard to describe binary numbers: '111'. The number of digits following the b may vary and, as do bit strings, may have a space every 4 digits.

1.2.2 Hex Notation

This standard uses a "C-like" notational convention to describe hexadecimal numbers, e.g. "0x36".

1.2.3 Equals and Assignment

The assignment operator "=" and the Equal To relational operator "==" are freely mixed in this standard. As in PASCAL or BASIC programming languages, the single equal symbol can be used to mean, "is equal to", in addition to "is assigned the value". The text may also use the equal symbol to mean, "evaluates to the value of", as in: "864/75 = 11.52". The interpretation of the single equal symbol within this standard depends on context and the most logical reading of the text.

1.2.4 Bitstream Syntax Notation

Where fields have the individual bits numbered, the lowest numbered bit in a multi-bit numbered value is the least significant bit.

In some syntactical structures mnemonics are used to define the order of bits in that field. The mnemonic "bslbf" is defined to be bit string, left bit first, where left-to-right is the order in which bit strings are written.

NOTE—Bit strings are written as a string of 1s and 0s within single quote marks, e.g. '1000 0001'. Blanks within a bit string are typically every four bits for ease of reading and have no significance.

1.2.5 Acronyms & Mnemonics

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Exchange
ATSC	Advanced Television Systems Committee
ATV	Advanced Television
bg	Background
bps	Bits per Second
BOC	Beginning of Command
Bps	Bytes per Second
bs	Block Size
BS	Back Space
bslbf	Bit String Left Bit First
CC	Closed Caption
cc	Column Count
CCP	Caption Channel Packet
CEA	Consumer Electronics Association
cl	Column Lock
CLW	Clear Window
COC	Continuation of Command
CR	Carriage Return [Note: This is not the code group CR]
CWx	Set Current Window
DER	Delete to End of Row
DFx	Define Window
DLC	Delay Cancel
DLW	Delete Windows
DLY	Delay
DSW	Display Windows
DTV	Digital Television
DTVCC	Digital Television Closed Caption
EG	Engineering Guideline
EIA	Electronic Industries Alliance
EIT	Event Information table
ETX	End of Text

FCC	Federal Communication Commission
fg	Foreground
FIFO	First In First Out
fps	Frames per Second
FF	Form Feed
GOP	Group of Pictures
GUI	Graphical User Interface
HCR	Horizontal Carriage Return
HDW	Hide Windows
HDTV	High Definition Television
ID	Identification
IEC	International Electrotechnical Commission
ISO	International Standards Organization
LCD	Liquid Crystal Display
MPEG	Motion Picture Experts Group
MS	Microsoft
NBS	Non-Breaking Space
NBTSP	Non-Breaking Transparent Space
NTSC	National Television Systems Committee
NUL	Null
OSI	Open Systems Interconnection
PMT	Program Map table
PSIP	Program and System Information Protocol
RP	Recommended Practice
rc	Row Count
RF	Radio Frequency
RGB	Red Green Blue
RI	Row Lock
RST	Reset
SDTV	Standard Definition Television
SMPTE	Society of Motion Picture and Television Engineers
SP	Space
SPA	Set Pen Attributes
SPC	Set Pen Color
SPL	Set Pen Location
STB	Set Top Box
SWA	Set Window Attributes
TGW	Toggle Windows
TSP	Transparent Space
uimsbf	Unsigned Integer Most Significant Bit First
VBI	Vertical Blanking Interval
VBV	Video Buffering Verifier
Ww	WordWrap
XDS	eXtended Data Services

1.2.6 Terms and Definitions

padding - bytes delivered in the DTVCC Caption Channel present to maintain the desired bandwidth of the caption channel in the digital multiplex. Padding bytes are not valid DTV captioning data,

syntactic element – See the second paragraph of Section 7.

syntax element – a field in a data structure.

CEA-608 datastream – a sequence of bytes conforming to CEA-608 [1] that includes captions, text, and XDS.

2 Normative References

The following references contain provisions, which, through reference in this text, constitute normative provisions of this standard. At the time of publication, the edition indicated was valid. All standards are

subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below.

2.1 Normative Reference List

- [1] CEA-608-E, Line 21 Data Service, March, 2008.
- [2] ISO/IEC 8859-1, Information technology - 8-bit single-byte coded graphic character sets—Part 1: Latin alphabet No. 1 (4/15/1998)
- [3] SMPTE RP 218-2002, Specifications for Safe Action and Safe Title Areas for Television Systems

2.1.1 Normative Reference Acquisition

ANSI or CEA Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com>; Email global@ihs.com

ISO/IEC Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com>; Email global@ihs.com
- IEC Central Office, 3, rue de Varembe, PO Box 131, CH-1211 Geneva 20, Switzerland; Phone +41 22 919 02 11; Fax +41 22 919 03 00; Internet <http://www.iec.ch>; Email pubinfor@iec.ch
- ISO Central Secretariat, 1, rue de Varembe, Case postale 56, CH-1211 Geneve 20, Switzerland; Phone +41 22 749 01 11; Fax +41 22 749 01 55; Internet <http://www.iso.ch>; Email mbinfo@iso.ch

SMPTE Standards:

- Society of Motion Picture & Television Engineers, 3 Barker Ave., White Plains, NY, 10601 USA
Phone: 914-761-1100 Fax: 914-761-3115, Internet <http://www.smpte.org>; Email: eng@smpte.org

2.2 Informative References

The following references contain provisions, which, through reference in this text, constitute informative provisions of this standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below.

2.2.1 Informative Reference List

- [1] ANSI INCITS 4, Information Systems—Coded Character Sets—7 Bit American National Standard Code for Information Interchange (7-BIT ASCII), 2002
- [2] ATSC A/53 Part 4, MPEG-2 Video System Characteristics
- [3] ATSC A/65, Program and System Information Protocol for Terrestrial Broadcast and Cable
- [4] ATSC A/153, ATSC Mobile Digital Television System, Advanced Television Systems Committee, Washington, DC.
- [5] ISO 639-2, Codes for the representation of names of languages—Part 2: Alpha-3 code as maintained by the ISO 639/Joint Advisory Committee (ISO 639/JAC),
<http://www.loc.gov/standards/iso639-2/iso639jac.html>; ISO 639-2 standard online:
<http://www.loc.gov/standards/iso639-2/langhome.html>.
- [6] ISO/IEC IS 13818-1:2007 (E), International Standard, Information technology - Generic coding of moving pictures and associated audio information: systems
- [7] ISO/IEC IS 13818-2:2000 (E), International Standard, Information technology—Generic coding of moving pictures and associated audio information: video
- [8] ISO/IEC 2022, Information technology - Character code structure and extension techniques (4/15/1999)
- [9] SCTE 21, Standard for Carriage Of NTSC VBI Data in Cable Digital Transport Streams

- [10] SMPTE Standard 12M, Television, Audio and File—Time and Control Code (1999)
- [11] SMPTE EG 43-, System Implementation of CEA-708-B and CEA-608-B Closed Captioning
- [12] SMPTE 333, DTV Closed-Caption Server to Encoder Interface (Revision of SMPTE 333M-1999)
- [13] SMPTE 334-1, Vertical Ancillary Data Mapping of Caption Data and Other Related Data
- [14] SMPTE 334-2, Caption Distribution Packet (CDP) Definition
- [15] SMPTE RP 2007, Closed-Caption CDP and "Grand Alliance" Serial Interfaces for DTV

2.2.2 Informative Reference Acquisition

ATSC Standards:

- Advanced Television Systems Committee (ATSC), 1776 K Street N.W., Suite 200, Washington, DC 20006; Phone 202-872-9160; Internet <http://www.atsc.org>

SMPTE Standards:

- Society of Motion Picture & Television Engineers, 595 W. Hartsdale Ave., White Plains, NY 10607-1824 USA Phone: 914-761-1100 Fax: 914-761-3115, Email: eng@smpte.org; Web: <http://www.smpte.org>

ISO Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com>; Email global@ihs.com
- ISO Central Secretariat Office, 1, rue de Varembe, Case postale 56, CH-1211 Geneve 20, Switzerland; Phone +41 22 749 01 11; Fax +41 22 733 34 30; Internet <http://www.iso.ch>; Email central@iso.ch

2.3 Compliance Notation

As used in this document “shall” and “must” denote mandatory provisions of the standard. “Should” denotes a provision that is recommended but not mandatory. “May” denotes a feature whose presence does not preclude compliance, and implementation of which is optional. “Optional” denotes items that may or may not be present in a compliant device.

3 Caption Channel Layered Protocol

A formal data communications channel protocol has been established for the DTVCC caption data channel. This formalization provides a framework for describing the caption communications hierarchy. Grouping the structures, concepts and features of this environment into the following hierarchical layers, aids in the understanding of the organizational aspects of the DTVCC system.

There are 5 layers in the DTVCC data framework (see Figure 1):

- DTVCC Transport Layer,
- DTVCC Packet Layer,
- DTVCC Service Layer,
- DTVCC Coding Layer, and
- DTVCC Interpretation Layer.

Similar to the OSI Reference Model, each of these layers provides particular services in the receiver, as shown in Table 1.

Services in a Receiver	Section	DTVCC Protocol Model	CEA-608 datastream
• Specification of Window appearance and content • Placement of Windows over Video • Synchronization of Command Interpretation to the Display of video • Reset of Caption Service	8	DTVCC Interpretation Layer	These layers are defined in CEA-608 [1].
• Parsing of Syntax	7	DTVCC Coding Layer	
• Demultiplexing of Caption Services	6	DTVCC Service Layer	
• Detection of Synchronization Loss • Resynchronization of Data Stream • Reset of Caption Decoder	5	DTVCC Packet Layer	
• Extraction of closed caption data from parent transport system, e.g. MPEG user data • Demultiplexing of CEA-608 datastream and DTVCC caption streams	4	DTVCC Transport Layer	

Table 1 DTVCC Protocol Stack

The DTVCC Transport Layer is where DTVCC data leave the DTV Video subsystem and are introduced to the DTV Closed-Caption decoder. Within the DTVCC decoder, DTVCC data are further processed up through the remaining layers of the DTVCC Protocol Model. Both the CEA-608 datastream and DTVCC data are carried in this layer.

NOTE—The DTVCC Transport layer should not be confused with the main system transport layer. For example, in MPEG-2 ATSC systems, the MPEG-2 transport layer is defined in

ISO/IEC 13818-1 [6], and this transport layer carries video, audio, data and control streams. DTVCC bits are encoded within the video signal conforming with ISO/IEC 13818-2 [7]. These bits enter the DTVCC decoder from the video stream's user data structure.

This standard specifies a method of placing a CEA-608 datastream, embedded in the signal specified by this standard, into the correct bit positions and fields in an NTSC signal, but this standard does not specify any application or presentation level processes for CEA-608 datastreams. Further processing of CEA-608 datastreams is specified in CEA-608 [1].

The DTVCC Packet Layer marks where DTVCC data enter the DTVCC decoder. This is a protocol data reassembly layer that buffers incremental bitstream data into a byte-aligned, multi-byte packet. This layer provides resynchronization for decoding the DTVCC stream, and is not dependent on the DTVCC Transport layer.

Processing of the Caption Channel packet data begins in the DTVCC Service Layer. Caption Channel packets are broken up into the encapsulated sub-blocks of data to be routed to one or more separate caption service processing routines within the decoder.

Using information in the DTVCC Service Directory (see Section 4.5), TV viewers may choose to view the processed data for one or more DTVCC services at a time. For example, a caption channel may contain an English language service and a Spanish language service.

The DTVCC Coding Layer breaks out the individual caption commands and caption text sequences from the service data blocks.

The DTVCC Interpretation Layer processes the caption elements presented by the DTVCC Coding Layer.

The DTVCC Interpretation Layer also provides for delaying the interpretation of the command stream and reset of a specific Caption Service.

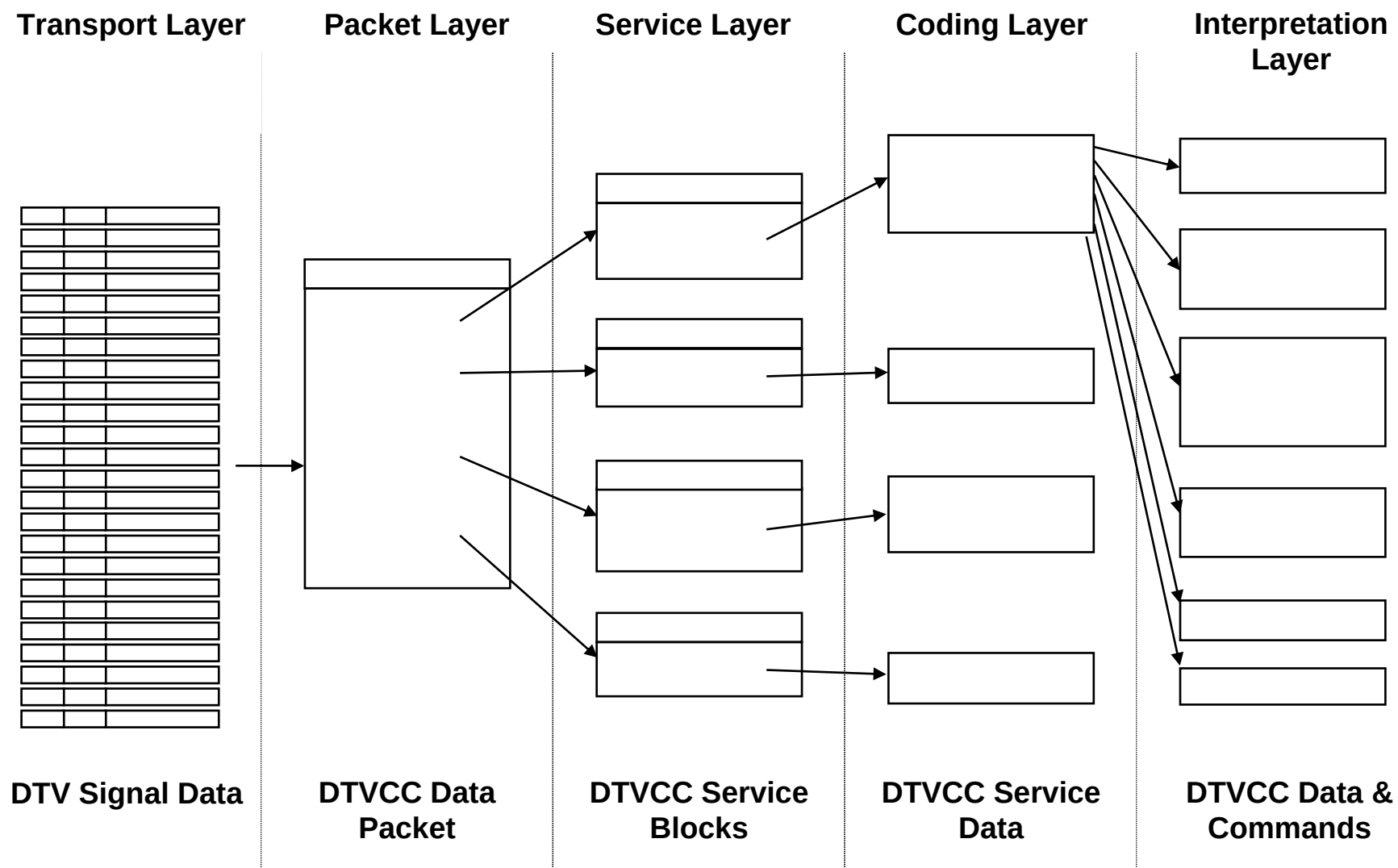


Figure 1 DTV Closed-Captioning Protocol Model

4 DTVCC Transport Layer

Section 4 describes the transport layer requirements for closed caption information within the DTV Bitstream, carriage of the CEA-608 datastream and DTVCC Caption Channel over the DTVCC Transport Channel, and carriage and semantic requirements of the DTVCC Service Directory information (see Section 4.5)

The 9600 bit per second Transport Channel described in Section 4 and subsections shall be a fixed-allocated 9600 bps rate, unless otherwise specified by referencing standards, e.g. ATSC A/153 [4].

4.1 DTVCC Transport Channel Data Rate

The DTVCC design allows for caption data to be transmitted at various data rates within a DTV video signal. However, it is often a continuous 9600 bits per second (bps) stream allocated from the DTV video signal capacity.

NOTE—This value of 9600 bps is divided by 1.001 when the frame rate is 23.98, 29.97, or 59.94 Hz.

In order to provide such a continuous stream and to ensure that specific caption data will reach receivers as intended (in relationship to the audio and video), this data channel is allocated on a frame-by-frame basis such that 1200 bytes of data (see Note above) are transported per second. For example, for a DTV video signal with fixed-allocated 9600 bps (1200 bytes per second) captioning, a 60 Hz frame rate and no CEA-608 captions, 20 bytes of DTVCC caption data are allocated in each frame.

The allocation for CEA-608 closed-caption encoding within the DTVCC Transport Channel shall be included in the allocated DTVCC Caption Channel bandwidth. That is, when the total DTVCC Transport Channel (consisting of the DTVCC Caption Channel and the CEA-608 Caption Channel) is 9600 bps, on average, a CEA-608 datastream is allocated 960/1.001 bps, and DTVCC captions are allocated at least 8640/1.001 bps.

4.2 Pre-Allocated Bandwidth

The DTVCC Transport Channel is typically a fixed, pre-allocated stream which exists in all DTV-system bitstreams, even though caption data may not be present. This ever-present bandwidth (which includes CEA-608 and DTVCC caption data) allows encoders to easily insert caption data into the DTV bit-stream at the point of origin and at multiple down-stream encoding points without having to perform complex data processing and bandwidth re-allocation.

The requirement to provide the DTVCC Transport Channel in a digital television stream is not part of this standard.

DTV processing systems extract these DTVCC bytes from this high rate transport stream, using parsing and buffering techniques outside the scope of this standard, to deliver the DTVCC content to a DTVCC processing subsystem at the typical 9600 bps bandwidth. Accordingly, the DTVCC processing subsystems shall be able to accommodate the 9600 bps rate.

4.3 DTV cc_data() Syntax and Semantics

Section 4.3 defines syntax and semantics of the **cc_data()** structure, and specifies rules for placing caption information into the **cc_data()** structure and interpreting that data. See Section 4.3.7 and Annex C for examples. See Annex G for an illustration of the structure of **cc_data()** and the closed caption data structures carried within.

Syntax	No. of Bits	Format
cc_data() {		
reserved	1	'1'
process_cc_data_flag	1	bslbf
zero_bit	1	'0'
cc_count	5	uimsbf
reserved	8	'1111 1111'
for (i=0 ; i < cc_count ; i++) {		
one_bit	1	'1'
reserved	4	'1111'
cc_valid	1	bslbf
cc_type	2	bslbf
cc_data_1	8	bslbf
cc_data_2	8	bslbf
}		
}		

Table 2 cc_data() Syntax

process_cc_data_flag—This one-bit flag shall indicate the validity of the remaining fields of the **cc_data()** structure. If it is set to '1', the **cc_data()** structure contains valid closed caption data. When it is set to '0', the **cc_data()** structure does not contain valid closed caption data. Decoders are expected to discard the contents of the **cc_data()** structure when **process_cc_data_flag** is set to '0'.

zero_bit—This bit shall be set to '0' to maintain backwards compatibility with previous versions of this structure.

cc_count—This 5-bit unsigned integer in the range 0 through 31 shall indicate the number of iterations of the "for" loop following this field. Sixteen (16) bits of closed caption payload data are carried in each pair of the syntax elements **cc_data_1** and **cc_data_2**. See Section 4.3.6.

one_bit—This bit shall be set to '1' to maintain backwards compatibility with previous versions of this structure.

cc_valid—This one-bit flag shall indicate the validity of the two closed-caption data bytes which follow it. If **cc_valid** is set to '1', then the following two bytes of closed-caption data shall be interpreted according to Section 5 through Section 9. If **cc_valid** is set to '0', the following two closed caption bytes shall have no meaning; however, the bit pair **cc_type** shall have meaning.

For CEA-608 captions, **cc_valid** shall be interpreted at the time it is inserted into the NTSC video. See Section 4.3.2.2.

cc_type—This two-bit field shall signal the type of the following two bytes of closed caption data as indicated in Table 3.

cc_type	cc_valid	cc_data_1 and cc_data_2
00	0	CEA-608 line 21 field 1 – do not generate run-in clock and start bit; or DTVCC padding bytes*
	1	CEA-608 line 21 field 1 closed caption bytes
01	0	CEA-608 line 21 field 2 – do not generate run-in clock and start bit; or DTVCC padding bytes*
	1	CEA-608 line 21 field 2 closed caption bytes
10	0	DTVCC padding bytes
	1	CCP Data – cc_data_1 and cc_data_2 are CCP Data, (Figure 4)
11	0	DTVCC padding bytes
	1	Start of CCP– cc_data_1 is a CCP Header byte, cc_data_2 is CCP Data (Figure 4)
* Depends upon context; see Section 4.3.1 and Figure 2		

Table 3 Closed-Caption Type (cc_type) Coding

cc_data_1—The first of two data bytes.

cc_data_2 – The second of two data bytes. Together these form a byte pair.

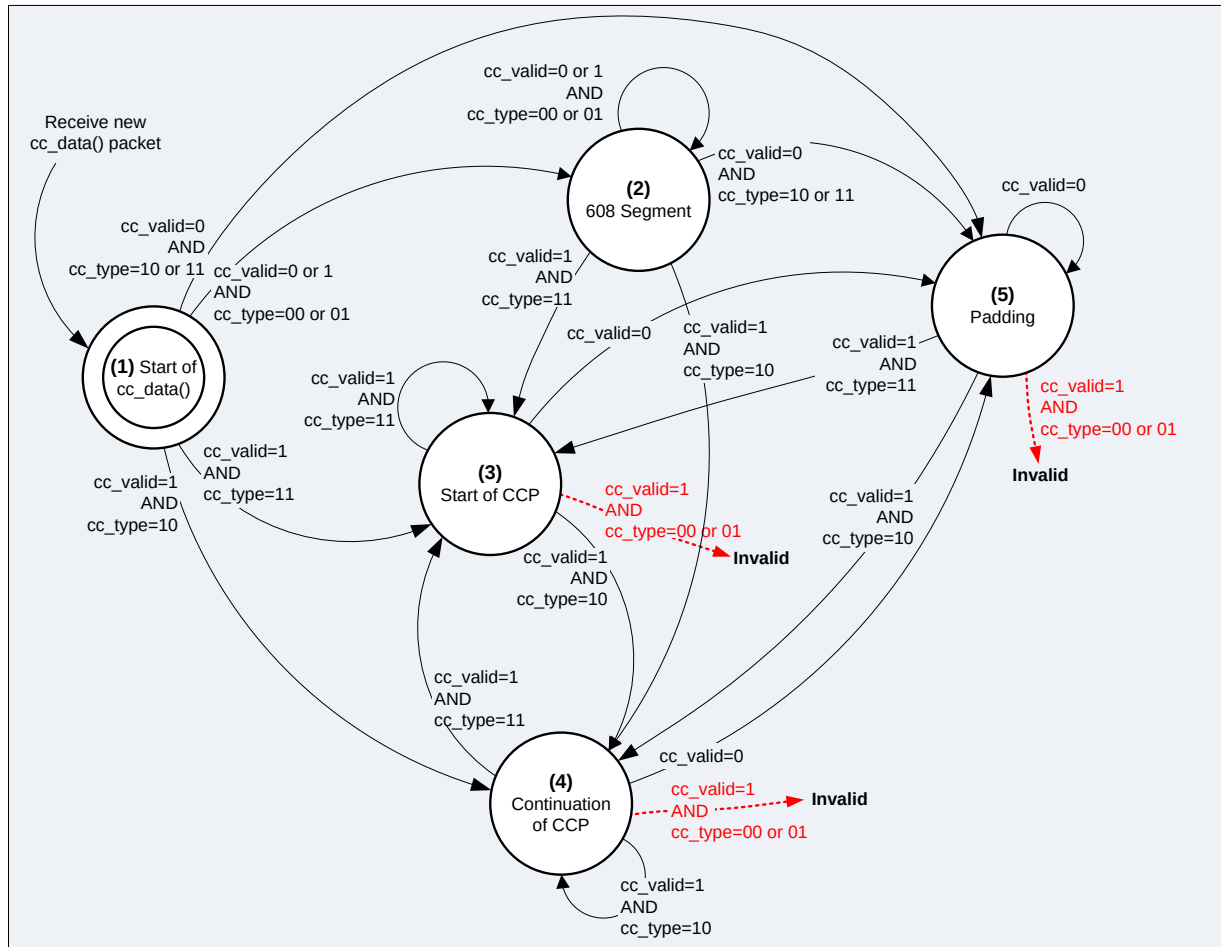
reserved – Syntax elements marked “reserved” are not currently defined, have no meaning, and shall be encoded with default values. Future versions of this standard may use other values for these reserved syntax elements; therefore decoders shall ignore these reserved syntax elements, but interpret portions of the syntax that are defined.

4.3.1 Structure and Interpretation of cc_data()

As shown in Table 3, when **cc_valid**=1, the **cc_type** field identifies the accompanying byte pair **cc_data_1** and **cc_data_2** as one of four items: CEA-608 line 21 field 1 or field 2 caption data, the start of a DTVCC Caption Channel Packet, or the continuation of a DTVCC Caption Channel Packet. When **cc_valid**=0 and **cc_type**=10 or 11, the byte pair is padding.

The state diagram in Figure 2 illustrates the context-dependent meaning of **cc_data** bytes carried in the **cc_data()** structure. State 1 (Start of **cc_data()**) is entered at the beginning of each instance of a new **cc_data()** structure. The formation of **cc_data()** conforms to the following rules:

1. When data of **cc_type** 00 or 01 occur at the beginning of the **cc_data()** structure, **cc_data_1** and **cc_data_2** shall contain CEA-608 data. The presence of CEA-608 data shall correspond with State 2 in Figure 2.
2. CEA-608 data bytes shall only occur in the first one, two or three byte-pair positions, depending on frame rate. Section 4.3.2 describes how this data may be stored in the field buffer model. The CEA-608 data shall follow the format and protocol as specified in CEA-608. The CEA-608 data field order (field 1 vs. field 2) depends on the field display order.
3. Data bytes **cc_data_1** and **cc_data_2** may be DTVCC padding bytes. The interpretation of **cc_data_1** and **cc_data_2** as DTVCC padding shall be as given in Figure 2; e.g. the conditions corresponding to State (5).
4. The condition **cc_valid** = '1' and **cc_type** = '11' shall indicate start of CCP (State (3)).
5. The condition **cc_valid** = '1' and **cc_type** = '10' shall indicate continuation of a CCP (State (4)).
6. The DTVCC emission shall structure **cc_data()** to conform only to valid Figure 2 state transitions. The condition **cc_valid** = '1' and (**cc_type** = '00' or **cc_type** = '01') occurring at states 3, 4, or 5 shall be considered invalid. Decoders may respond to such conditions by displaying unexpected information and the decoder may be left in an indeterminate state.



4.3.2.1 Transport

When an NTSC program is compressed as DTV for transmission in a DTV system, the entire CEA-608 datastream (i.e. captions, text, and eXtended Data Service (XDS)) should be preserved during the compression process (except for appropriate changes to locally generated XDS packets).

4.3.2.2 Generating a CEA-608 Waveform

This section describes a model for the generation of a CEA-608 waveform. This model describes decoder operation and requirements for streams compliant with this standard; however, receiver implementation may use any process providing equivalent functionality when receiving compliant streams.

NOTE—This model turns off the run in clock; however, there is no requirement that the run in clock ever be turned off.

Two FIFO-based field buffers (one for **NTSC_Line21_field1_data** and one for **NTSC_Line21_field2_data**), as shown in Figure 3, hold byte-pairs from the CEA-608 datastream and associated **cc_valid** flags.

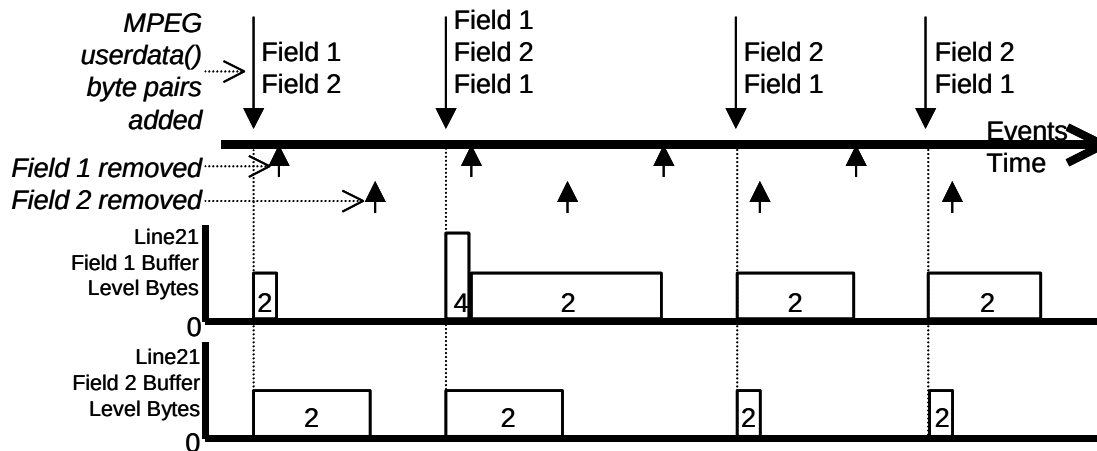


Figure 3 Example of CEA-608 Captioning Field Buffers

Byte pairs are received from the DTVCC Transport and written into the field buffers in the order that they appear in the **cc_data()** structure.

Bytes are removed from field buffer “x” at the start of NTSC field “x”, and are transmitted on the NTSC signal in the order they are removed from the buffer.

If field “x” Buffer is empty at the transmit time of NTSC field “x”, a CEA-608 waveform should be generated for that field with **cc_data_1** = 0x80 and **cc_data_2** = 0x80 (in CEA-608 notation, two 0x00s with odd parity).

If the byte pair at the output of the FIFO is marked **cc_valid** = 0, no CEA-608 waveform shall be generated for field “x,” except that if captions are generated for field 2, CEA-608 requires that clock run in and start bits shall also be generated for field 1.

If a video frame has no associated **cc_data()** structure or if **process_cc_data_flag** = 0, no byte pairs shall be added to the field buffers.

In steady-state operation, the proper field buffer always contains data at the start of a field for all frame rates and repeated fields. Designers should understand that the use of frame rates of 60, 30 or 24 Hz may cause corrupted CEA-608 caption sequences at start-up and across splices.

CEA-608 [1] requires that a field 1 waveform (run-in clock and start bit) be present whenever a field 2 waveform is present.

Encoding equipment shall not assume that decoders can buffer more than two byte-pairs (four bytes) of CEA-608 field 1 data and two byte-pairs (four bytes) of CEA-608 field 2 data, but shall send a CEA-608 datastream at a substantially constant data rate. The size of the field buffers shall be at least two byte-pairs for each type, since some **cc_data()** structures can have two byte-pairs of one type in them. Therefore, encoders may not assume a larger value when arranging the placement of a CEA-608 data in the **cc_data()** structure.

Each field's buffer receives zero, two, or four bytes synchronized to the video frame, and two bytes are removed at the encoding time of the NTSC output. If there are exactly 1000 byte pairs of a type per 1001 video frames, then, under steady-state operation, the buffer always has two bytes available for each NTSC output.

4.3.2.2.1 Alternating Transmission of CEA-608 Datastream

When CEA-608 data is present in a stream, encoding equipment shall always include a **cc_data()** structure for each field, even if a decoder is being instructed not to generate the run-in clock and start bit on one field of CEA-608 data (by setting **cc_valid** to 0 for that field type).

An exception occurs when the decoded frames produce a frame rate higher than 30/1.001, for example, when the frame rate is exactly 30 Hz. In this case, frames with a reduced number of **cc_type** = 00 or = 01 at the beginning of the **cc_data()** structure may occur to lower the average rate to no more than 30/1.001 byte pairs of each field type per second.

This allows downstream equipment to insert a legal CEA-608 datastream without moving DTVCC caption data between video frames.

4.3.2.2.2 Suppressing Run-In Clock and Start Bit When No CEA-608 Data Present

When decoders do not have CEA-608 **cc_data_1** and **cc_data_2** bytes to encode into NTSC outputs, decoders may suppress the run-in clock and start bit on the NTSC output signal, but there is no requirement that they do so. Likewise, if the entire **cc_data()** structure is missing from the video frame, decoders may suppress the run-in clock and start bit on the NTSC output signal, but there is no requirement that they do so.

4.3.2.2.3 Data Rate Reduction Considerations

Data rate reduction can be done in an encoder by periodically sending **cc_data()** structures without **cc_type** = 00 or 01. See Table 3. The decoder does not buffer any data when decoding this frame's data. This would normally produce a "captionless" frame (no captioning waveform on line 21). If captionless frames occur every 1001 frames, then, after startup, the field buffer does not empty (assuming the decoder is locked to the DTVCC Transport timebase).

The number of frames with a reduced **cc_type** = 00 or = 01 count at the beginning of the **cc_data()** structure should be minimized, but may be more frequent in portions of the program where splices have occurred. These reduced caption frequency portions of the sequence may create NTSC output with corrupted or missing caption data during steady-state operation

An NTSC frame could be encoded without CEA-608 data at start-up, if a **cc_data()** structure with a reduced CEA-608 datastream fails to fill the buffer before an NTSC frame is output. This could cause corruption in the NTSC output signal, since CEA-608 decoding rules include special cases for repeated control codes in adjacent frames. Insertion of an empty line 21 could change these to non-adjacent control codes, changing their interpretation. This is unavoidable and anticipated for FCC compliant CEA-608 caption decoders.

4.3.3 DTVCC Caption Channel Packets in cc_data()

DTV Closed-Caption data is transmitted (encoded) in variable-rate, variable-sized DTVCC Caption Channel Packets (CCPs) (see Section 5). To allow for simple extraction and insertion of these packets

without the need to fully parse their contents, the first byte-pair of each CCP is marked using **cc_type** = 11 (i.e., CCP start). The remaining byte-pairs of the DTVCC packet are identified with **cc_type** = 10 (i.e., CCP data).

A CCP may continue from one **cc_data()** structure to the next. The end of a CCP is indicated by either:

- (1) receipt of a subsequent CCP header (**cc_valid** = 1, **cc_type** = 11)
- (2) processing the number of bytes indicated by **packet_size_code** (see Section 5).

If a single DTVCC Caption Channel Packet is to be spread across several **cc_data()** structures, at least one byte pair should be included in each **cc_data()** structure.

If at least one byte pair of the DTVCC Caption Channel Packet is included in the **cc_data()** structure, the rest can be left unused by padding the end of the **cc_data()** structure. See Annex C.

Caption Channel Packets may be intentionally shortened; decoders should not assume a reset operation or other error recovery operation when CCPs shorter than the expected length are found in the data stream.

If a syntactic element is partially received when the end of a CCP is found, the syntactic element shall be discarded.

NOTE—In most video compression standards, coded frames are transmitted in a different order (“Transport” order) than they are displayed (“Display” order). Since the DTVCC captioning data extensions are often part of the video coded frame constructs and follow the video compression coded-picture reordering process, the order in which the **cc_data()** structure is transmitted might not be the same order in which they will be processed by the DTVCC Decoder. The DTVCC Decoder reorders **cc_data()** structures along with the frames to which they correspond prior to the DTVCC Caption Channel Packet location and extraction method described previously. Once these captioning data are reordered, they are ready for processing by the DTVCC Decoder. See Annex D.

4.3.4 Caption Order in Progressive and Interlaced Frames

For progressive frames, the order of CEA-608 captions is only partially specified. This is because the conversion from a progressive frame to interlaced NTSC frames can be done producing a field 1 followed by a field 2, or vice versa.

4.3.5 Padding Unused Space in **cc_data()**

The value of **cc_type** is important, even if **cc_valid** = 0.

- At the beginning of a **cc_data()** structure, byte pairs with **cc_valid** = 0 and **cc_type** = 00 or 01 are not padding, but signify that no CEA-608 waveform is to be generated for the associated field (see section 4.3.2.2).
- A padding byte pair with **cc_valid** = 0 and **cc_type** = 11 or 10 signals the end of the CEA-608 data within the **cc_data()** structure (This is depicted in Figure 2)

It is recommended that the padding byte pairs use values **cc_data_1** = 0x00, **cc_data_2** = 0x00. The integrity of data values associated with **cc_valid** = 0 is not guaranteed.

4.3.6 Frame Rates

The value of **cc_count** varies depending on the frame rate and coding mode.

Video encoding standards and the standards that reference them specify the methods whereby caption essence is transported within the video syntax.

Each location within the video syntax in which closed caption data is legally allowed to appear is considered an “opportunity.” For systems in which a fixed-allocated bandwidth for closed caption data is required, the **cc_data()** structure shall be included within the video syntax at every available opportunity. For such systems, the value of **cc_count** in each instance of **cc_data()** is set based on the difference

between the presentation time of the picture containing the **cc_data()** and the presentation time of the picture carrying the **cc_data()** in the immediately preceding picture in the presentation timeline.

Given:

r = the desired average rate of cc_data_1 and cc_data_2 in bits per second

t = the time in milliseconds since the last opportunity to insert caption data

then the **cc_count** value can be computed as follows:

$$cc_count = \frac{(r \times t)}{2} \times \left(\frac{1 \text{ byte}}{8 \text{ bits}} \right) \times \left(\frac{1 \text{ sec.}}{1000 \text{ msec.}} \right) = \frac{r \times t}{16,000}$$

Note that for non-integer frame rates such as 29.97 fps, this equation will yield a non-integer result. In such cases, the result shall be rounded to the nearest whole value.

The rate of delivery of the CEA-608 datastream shall not exceed the data rate of (60/1.001 * 2) bytes per second. This data rate requirement shall be met across splices. For example, when two sequences with **frame_rate_value** of either 30 or 60 are spliced, the splicer could delete CEA-608 caption information from either the last frame of the ending sequence or the first frame of the new sequence.

4.3.6.1 Examples (Informative)

As one example of the “opportunity” concept, for the ATSC standard using MPEG-2 video encoding, as specified in Section 6.2.1of A/53 Part 3 [6], closed caption data is included in the video syntax in **extension_and_user_data(2)** at the picture level. Each occurrence of a location in the video syntax where **extension_and_user_data(2)** is legally allowed to appear represents an opportunity to place **cc_data()**.

Table 4 below shows the relationship between the number of DTV data bytes vs. CEA-608-E bytes contained in the **cc_data()** structure.

Coded Video Format	cc_count		No. CEA-608 bytes cc_type=00 or 01 cc_data_1 and cc_data_2	No. DTVCC bytes cc_type=10 or 11 cc_data_1 and cc_data_2
24p	25	Alternating between	4	46
			6	44
30i	20		4	36
30i with 3:2 pull-down	Alternating between	20	4	36
		30	6	54
30p	20		4	36
60p	10		2	18

Typical Cases: Field 1 and Field 2 CEA-608 Data Present

Table 4 DTVCC Example #1 - MPEG-2 Video Transport Channel—cc_data() parameters

Coded Video Format	cc_count	CEA-608 field 1 data present	CEA-608 field 2 data present	No. CEA-608 bytes cc_type=00 or 01 cc_data_1 and cc_data_2		No. DTVCC bytes cc_type=10 or 11 cc_data_1 and cc_data_2
24p	25	no	no	0		50
		yes	no	Alternating between	2	48
					4	46
30i	20	no	no	0		40
		yes	no	2		38
30i with 3:2 pull-down	Alternating between	20	no	0		40
			yes	2		38
	30	no	no	0		60
				Alternating between	2	58
		yes	no		4	56
30p	20	no	no	0		40
		yes	no	2		38
60p	10	no	no	0		20
		yes	no	Alternating between	0	20
					2	18

Example Cases where Field 1 and Field 2 CEA-608 Data Are Not Both Present

Table 5 DTVCC Example #2 - MPEG-2 Video Transport Channel—cc_data() parameters

The tables above show typical values for **cc_count** for various video codings. It should be noted that the number of bytes at each instance in the stream may not be constant. This is evidenced by the two entries coded at 30i with different source video. “Alternating between” means that both values appear in the output sequence about half the time. This can be caused by a film 3:2 pull-down sequence. Similarly, dual **cc_data_1** and **cc_data_2** entries will use each value about half the time.

4.3.7 Example cc_data() Contents

Table 6 illustrates an example of the arrangement of the closed-caption data stream in the **cc_data()** structure for a typical 30 frames per second video signal frame where **cc_count** = 20. A Caption Channel Packet starts immediately after the CEA-608 byte pairs. Data between byte pairs 4 and 20 is obvious from context, the values of index ‘i’ are from Table 2.

i+1	cc_valid	cc_type	cc_data_1	cc_data_2
1	1	00	CEA-608 data, field 1	CEA-608 data, field 1
2	1	01	CEA-608 data, field 2	CEA-608 data, field 2
3	1	11	byte #1: CCP Header ²	byte #2: CCP Data
4	1	10	byte #3: CCP Data	byte #4: CCP Data
...				
20	1	10	byte #35: CCP Data	byte #36: CCP Data

Table 6 Aligned cc_data() structure and CCP Example

² CCP Header is defined in Figure 4, Section 5.1.

Table 7 shows the **cc_data()** structure encoding of unaligned CCPs. The end of a previous CCP (that was 128 bytes in length) is in the first 4 bytes of the **cc_data()** structure after the CEA-608 byte pairs. The start of the first 32 bytes of a new CCP fills the remainder of the 40 bytes of the **cc_data()** structure for a 30 frames per second video signal.

i+1	cc_valid	cc_type	cc_data_1	cc_data_2
1	1	01	CEA-608 data, field 2	CEA-608 data, field 2
2	1	00	CEA-608 data, field 1	CEA-608 data, field 1
3	1	10	byte #125: CCP data	byte #126: CCP Data
4	1	10	byte #127: CCP Data	byte #128: CCP Data
5	1	11	byte #1: CCP Header ³	byte #2: CCP Data
6	1	10	byte #3: CCP Data	byte #4: CCP Data
7	1	10	byte #5: CCP Data	byte #6: CCP Data
8	1	10	byte #7: CCP Data	byte #8: CCP Data
...				
20	1	10	byte #31: CCP Data	byte #32: CCP Data

Table 7 Unaligned Caption Channel Packet Example

The **marker_bits**, **cc_valid** and **cc_type** fields DO NOT use any part of the 9,600 bps bandwidth allocated for closed-caption data. Therefore, 9,600 bps will be maintained for the transport of closed-caption information, including the CEA-608 datastream and DTVCC captions, within the DTV stream.

If CCP data is present in the **cc_data()** structure, it is suggested that it be inserted at the beginning of the DTVCC data area, or at the end of the data structure, but not spread across the data structure. Table 8 illustrates a non-optimal, but valid syntax. In the Table 8 example, the first byte pair following the CEA-608 data (byte#41 and byte#42) are the end of a CCP started in a prior **cc_data()** structure,

i+1	cc_valid	cc_type	cc_data_1	cc_data_2
1	1	00	CEA-608 data, field 1	CEA-608 data, field 1
2	1	01	CEA-608 data, field 2	CEA-608 data, field 2
3	1	10	byte#41 = CCP Data	byte#42 = CCP Data
4	0	00	0xXX = padding	0xXX = padding
5	1	11	byte#1: CCP Header ⁴	byte#2: CCP Data
6	0	00	0xXX = padding	0xXX = padding
7	1	10	byte#3: CCP Data	byte#4: CCP Data
8	0	00	0xXX = padding ⁴	0xXX = padding
9	0	00	0xXX = padding	0xXX = padding
...				
20	1	10	0xXX = padding	0xXX = padding

Table 8 cc_data() Structure Example Showing Unusual Sequences of cc_valid

This is not recommended, but it is valid syntax. There may be reasons for the **cc_data()** structure to have interspersed groups of **cc_valid** = 0 byte pairs, such as the deletion of caption services in an edit of a DTVCC Transport stream. This example points out the possibility of unusual sequences of **cc_valid**.

4.4 Latency

As previously discussed in Section 4.3.2.2.3, the DTV frame data can be transmitted in “Transport Order”. This transmission path has an inherent latency associated with it, in that there can be a significant delay

³ CCP Header defined in Figure 4, Section 5.1.

⁴ The presence of padding should not be interpreted as terminating the CCP.

from the time the frame data enters a DTV encoder to the time these same data exit a DTV decoder and are available for display processing. The DTVCC data experience this same delay since they are transmitted within the **cc_data()** structure, even though the actual DTVCC data are introduced to the encoder in “display order”, as is the video.

4.4.1 Zero Delay Progressive Sequences

For non-interlaced (progressive) frames, decoders may produce an NTSC output with either field order. For example, if a video frame rate of 30 in a progressive sequence includes valid **cc_type** = 00 followed by **cc_type** = 01 in the **cc_data()** structure, then the decoder can produce an NTSC output with **cc_type** = 00 on the first (NTSC field 1) field derived from the progressive frame, and the **cc_type** = 01 on the second (NTSC field 2) field, or may put previous caption data on the first (NTSC field 2) field derived from the progressive frame and the **cc_type** = 00 data on the second (NTSC field 1) field. Either of these arrangements may be considered as “zero delay”.

NOTE—Reversal of the output order of **cc_type** 00 and 01 relative to the input order is not recommended, but is not explicitly forbidden. That is, the latency for **cc_type** 00 may differ from the latency of **cc_type** 01.

4.4.2 Zero Delay Interlaced Sequences

For interlaced frames, the **cc_type** 01 is associated with a Top Field, and **cc_type** = 00 with a Bottom Field. DTV decoders may translate Top Fields to NTSC field 1 or to NTSC field 2. If Top Fields are translated to field 1, then “zero delay” shall be defined as inserting the captioning data into the field immediately subsequent to the field in which it is transmitted. If Top Fields are translated to field 2, then “zero delay” shall be defined as inserting the captioning data into the field in which it was transmitted.

4.4.3 End-to-end Delay

The total delay for caption information relative to the video should be maintained at or below four NTSC frame times (that is, 4/30 second) for 60 Hz based signals and at or below four 25 Hz frames when associated with 50 Hz based frame rates. These delays include those of caption encoders, remultiplexers or other processing equipment, and decoders.

4.4.4 Encoding Delay

The delay for caption information relative to the video for encoders should be maintained at or below two NTSC frame times (that is, 2/30 second) for 60 Hz based signals and at or below two 25 Hz frames when associated with 50 Hz based frame rates.

4.4.5 Decoding Delay

The total delay for caption information relative to the video for decoders should be maintained at or below two NTSC frame times (that is, 2/30 second) for 60 Hz based signals and at or below two 25 Hz frames when associated with 50 Hz based frame rates.

4.4.6 Intermediate Processing Delay

Intermediate processors, including those that combine caption streams, re-multiplex video signals and otherwise process video independent of caption information should not introduce any systematic advance or delay of captions, unless the adjustment is introduced to compensate for a constant delay component elsewhere in the processing system.

4.5 Caption Service Metadata

Each caption service may have metadata items that describe the service. These metadata items may be carried “out of band” to the caption service data in the **cc_data()** structure such as elsewhere in the DTVCC Transport. The encoding syntax for metadata transmission are not within the scope of this standard.

The service metadata items are defined as follows:

NUMBER OF SERVICES: the total number of caption services (1-16) present over some transport-specified time period. There shall be a maximum of 16 separate services present.

NOTE—The signaling of this metadata need not apply to each **cc_data()** structure, but might be more useful in some applications as an aggregate of the caption services over some well-understood time period for the application, such as a program event duration.

For each caption service, the following metadata can be conveyed:

TYPE OF SERVICE: {608 | 708} the service contains a CEA-608 datastream or CEA-708 bytes. There shall be at most one CEA-608 datastream signaled. The CEA-608 data itself signals the individual CEA-608 caption channels. This is in contrast to CEA-708 services, which map 1:1 to the CEA-708 caption service numbers and additional metadata items.

When the TYPE OF SERVICE is 708, then the following 708-related metadata should be conveyed:

CAPTION CHANNEL SERVICE DIRECTORY: This is the aggregate of all metadata described below, for all services present over some period of time.

SERVICE NUMBER: the service number as found in the 708 caption service block header (1-31). This field provides the linkage of the remaining metadata to a specific 708 caption service.

LANGUAGE: the dominant language of the caption service. It is recommended that this be encoded from ISO 639-2 [5].

NOTE—An encoding is recommended to ensure an appropriate breadth of languages are unambiguously supportable. When a caption service contains multiple languages, this metadata item should indicate the predominant language most viewers would select.

DISPLAY ASPECT RATIO: {4:3 16:9} the display aspect ratio assumed by the caption authoring in formatting the caption windows and contents.

EASY READER: this metadata item, when present, indicates that the service contains text tailored to the needs of beginning readers.

4.5.1 Decoder Processing of Metadata

If metadata is available through multiple structures, these structures shall be consistent or have well defined priority rules. Acquiring all program information from a single, concise table facilitates fast tuning.

Decoders shall offer the user the ability to select any of the (up to sixteen) CEA-708 caption services. Decoder support for user selection of CEA-608 caption channels is optional.

Decoders that allow selection of CEA-608 caption services should allow any one of the four caption channels (CC1-CC4) to be selected, even if no metadata is present. This allows decoders to continue access to CC2 and CC4.

The presence or absence of CEA-608 metadata shall have no effect on the encoding of the CEA-608 data in the **cc_data()** structure onto NTSC output signals.

In the absence of metadata describing a caption service, the decoder should decode CEA-708 captions as if metadata was present with Display Aspect Ratio value of 16:9. In the event a decoder cannot render a caption window on the display device within the device's safe title area, the caption window can be completely disregarded. It is preferred that decoders transforming the aspect ratio of the received video (letterboxing a 16:9 image for 4:3 display, for example) apply similar transforms to received caption material to achieve captions coordinated with the displayed image format.

The grid locations for overlaying caption windows on an image are assumed to be from the content provider's perspective and refer to the transmitted frame. It is preferred that any image format alterations

performed in the decoder (for example, 'letterboxing') should be similarly applied to the DTVCC window positioning grid for the accompanying caption material.

4.6 Shortened Caption Channel Packets

Reception of a CCP Header prior to the reception of the number of bytes indicated in the **packet_size_code** value given in the prior CCP Header byte will terminate that CCP.

If a syntactic element is partially received when this condition occurs, it shall be discarded. This situation might arise when a long packet is being sent at the time of a splice; there is no way to continue the long packet into the new stream, so this shortening technique may be used.

4.7 Independent Control of Audio and Caption Services

The caption data shall be an independent service from the audio service. It is intended that hearing and deaf viewers may share the same living room, watching the same program. Each viewer should be able to experience the service that they use (audio or captions) independent of adjustment of the other; for example, the choice of audio language or caption language should be independently selectable. Devices using captioning data shall provide a mode in which no modification of audio settings (including language) occurs when captioning is enabled, changed or disabled. In addition, no modification of the captioning data display shall occur when audio settings (including language) are changed. However, other modes, such as 'caption when mute enabled' may be implemented.

5 DTVCC Packet Layer

The DTVCC Caption Channel data are framed as packets (CCPs) within the DTVCC Transport Channel prior to encoding. This packet structure is completely defined within this standard. (Error detection, error correction, compression, and other low-level transport overhead issues are handled in the DTV transmission layers, and thus fall outside the scope of this standard).

The DTVCC Packet Layer is defined by a variable-size CCP of **n** bytes of closed captioning data where **n** ≤ 128, and is an even number. CCPs are coded in the **cc_data()** structure of the video frames in the DTV stream. The beginning and ending of each CCP in the **cc_data()** structure is indicated by the syntax defined in Section 4.3.

5.1 Caption Channel Packet Syntax and Semantics

Each CCP consists of a one byte header and **n-1** bytes of data, where **n** is an even number indicating the total packet size. (See Figure 4.) The header contains the **packet_size_code** and **sequence_number**.

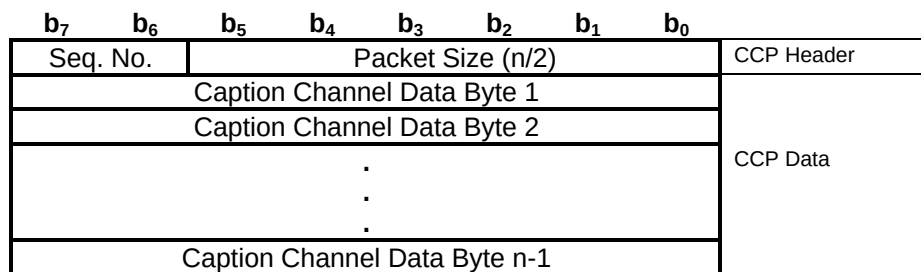


Figure 4 Caption Channel Packet

The bit-stream syntax for the **caption_channel_packet()** shall be as shown in Table 9.

	No. Of bits	Mnemonic
caption_channel_packet() {		
sequence_number	2	uimbsf
packet_size_code	6	uimbsf
for (i = 0; i < packet_data_size; i++) {		
packet_data[i]	8	bslbf
}		
}		

Table 9 DTVCC Caption Channel Packet Syntax

The variables, **sequence_number**, **packet_size_code**, **packet_data_size** and **packet_data** are defined as follows:

sequence_number is a two-bit (b6 - b7) rolling sequence (from 0 to 3) which may be used by receivers to detect the loss of packets. When packet loss is detected, any partially accumulated data from the previously received packet shall be discarded, and Reset command processing shall be performed for each existing Caption Service. See Section 8.9.5. Decoding shall resume with the first Service Block encountered in the new Caption Channel Packet. See Section 6.2.

Detection of data loss may be accomplished by using discontinuities in the **sequence_number** to detect loss of synchronization. Because there are only four possible values for **sequence_number**, not all data disruptions cause discontinuities in the sequence. For example, if the number of lost packets is a multiple of four, the count appears to be continuous, and packet loss is not detected.

Decoder designers may wish to implement an enhanced discontinuity detection method. At the start of each Caption Channel Packet, the command parser state is checked to assure that all syntactic elements from the previous packet have been completed, and no partially accumulated data remains. If data remains in the parser, then a discontinuity may be assumed to have occurred, since DTVCC Caption Channel Packets ought to only contain complete syntactic elements. See Section 6.3. The decoder would then perform Reset command processing for each Caption Service.

packet_size_code—Contained in the lower 6 bits of the header (b0 – b5), these bits define the number of byte pairs in the CCP including the header byte.

packet_data_size indicates the number of iterations of the For loop in the table above.

If (**packet_size_code** == 0), then **packet_data_size** = 127;

else **packet_data_size** = (**packet_size_code** * 2) - 1.

packet_data—This array of bytes is organized into Service Blocks, as described in Section 6.2. The processing of over-length packets is undefined. Encoder and processing equipment manufacturers should be aware that if the amount of Caption Channel Packet data is greater than indicated by **packet_size_code**, decoders may ignore the extra data.

5.2 CCP Processing

Figure 5 is state transition diagram that may be helpful in understanding the processing of CCPs. The references to “State 3” are to the state table in Figure 2.

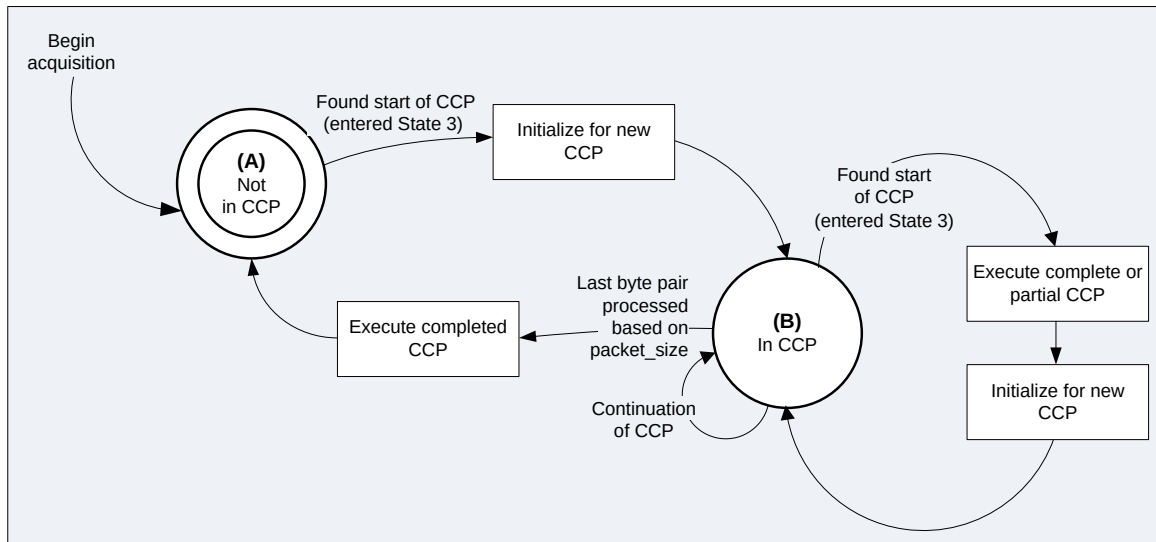


Figure 5 CCP State Table

As shown in Figure 5, a decoder enters state (A) when initially acquiring the caption service. When a **cc_data()** structure is processed in which the **cc_data()** state table (Figure 2) reaches State (3), the start of a CCP has been found, the receiver initializes its buffers for processing a new CCP, and state (B) is entered.

The end of the CCP is reached either by processing the last byte pair based on **packet_size_code**, or by the reception of another CCP start (re-entry to State (3)). When a CCP is completely received, or a partial CCP is received and the beginning of a new CCP cuts it short, the preceding complete or partial CCP is executed by the decoder.

6 DTVCC Service Layer

The DTVCC Caption Channel is divided into a set of logical sub-channels or "Services." The Service Layer defines the headers for the caption data channel Service Numbers, Service Types, and Service Attributes.

Receivers use the information in this layer to route the Service Data to the appropriate internal processing modules.

6.1 Services

The Caption Channel Services are of two types: 6 Standard services (Service #1 through Service #6) and 57 Extended services (Service #7 through Service #63) allowing for 63 total services. Service #0 shall not be used.

Service #1 is designated as the Primary Caption Service. This service contains the verbatim or near-verbatim, captions for the primary language being spoken in the accompanying program audio.

Service #2 is designated as the Secondary Language Service. This service contains captions in a secondary language, which are translations of the captions in the Primary Caption Service or may be captions of a Secondary audio service.

The other service sub-channels are not pre-assigned. It is up to the discretion of the individual caption provider to utilize the remaining service channels.

6.2 Service Blocks

Caption Channel Service Blocks provide the structure for the asynchronous time division multiplexing of Service data within the DTV Closed-Caption Channel. Service data are time-division-multiplexed into the appropriate time slots in the Caption Channel Packets as required. See Section 6.2.5.

Caption providers and encoding equipment algorithms govern the frequency, priority, and bandwidth consumption for each individual Service carried in these Service Blocks.

Each Service Block consists of a 1 or 2 byte header followed by 1 to 31 bytes of data. See Figure 6.

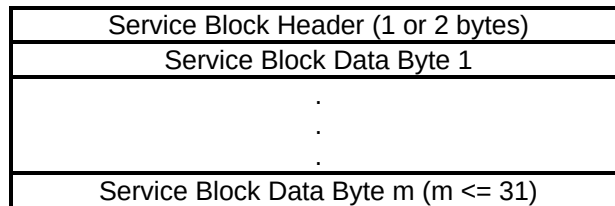


Figure 6 Service Block

The Service Block syntax is shown in Table 10 Service Block Syntax

	No. Of bits	Mnemonic
Service_block() {		
service_number	3	uimsbf
block_size	5	uimsbf
if (service_number == b'111' && block_size != 0){		
null_fill	2	'00'
extended_service_number	6	uimsbf
}		
if (service_number != 0) {		
for (i = 0; i < block_size; i++) {		
Block_data	8	bslbf
}		
}		
}		

Table 10 Service Block Syntax

6.2.1 Standard Service Block Header

A one-byte Standard Service Block Header consists of 2 parts: the Service Number (**sn**), and the Service Block Size (**bs**). See Figure 7. The Service Number is defined in the 3 high-order bits (**sn₀ - sn₂**), and the Service Block Size is defined in the 5 low-order bits (**bs₀ - bs₄**).

The service Block Size value ranges from 1 to 31 and indicates the number of bytes following the header. A Standard Service Block Size of 0 shall not be used, except in a Null Service block Header. See Section 6.2.3.

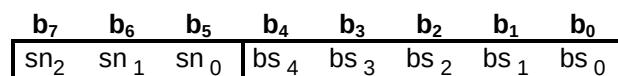


Figure 7 Service Block Header

6.2.2 Extended Service Block Header

If services numbered higher than 6 are employed, a 2-byte Extended Service Block Header is used, as shown in Figure 8. The first byte of this Extended Header has the same format as the Standard Service Block Header, with the exception that the 3 high order bits (**sn₂ – sn₀**) are each fixed to a value of 1. This value of “7” in the Standard Service Number bits signals the use of an Extended Service Block Header. The 2nd header byte contains an Extended Service Number (**sn₀ – sn₅**) whose value may range from 7 to 63. Extended service numbers less than 7 are not permitted, since they can already be specified by the Standard Service Block Header.

The Block Size value in the lower 5 bits (**bs₀ – bs₄**) of the first byte of the Extended Service Block Header signals the number of Service Block Data bytes that follow the Extended Service Block Header.

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
1	1	1	bs ₄	bs ₃	bs ₂	bs ₁	bs ₀
0	0	sn ₅	sn ₄	sn ₃	sn ₂	sn ₁	sn ₀

Figure 8 Extended Service Block Header

6.2.3 Null Service Block Header

The NULL Service Block Header is shown in Figure 9. All bits are set to 0. The appearance of a NULL Service Block Header indicates that there are no more Service Blocks in the Caption Channel Packet for the decoding hardware to process. This is equivalent to a Service Block with **service_number** == 0 and **block_size** == 0. When **service_number** == 0, **block_size** shall be 0.

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
0	0	0	0	0	0	0	0

Figure 9 Null Service Block Header

6.2.4 Service Block Data

Within each Service Block, the header may be followed by up to 31 bytes of Service Data. This data consists of Coded Caption Data and Data Interpretation. See Section 8.

Service Data are demultiplexed from the Service Blocks and routed to the appropriate service processors in the DTV receiver’s decoder. This separation creates the individual byte streams for each of the services, which are handed off to the DTVCC Coding Layer.

6.2.5 Service Blocks within Caption Channel Packets

Service Blocks are time division multiplexed and inserted sequentially into the Caption Channel Packets.

Service Blocks shall not cross Caption Channel Packet boundaries. If a service requires more data than the current Caption Channel Packet allows, the Service Block Data shall be truncated to fit the current packet. Any remaining data bytes shall be embedded in a new Service Block (with Header), and placed within a subsequent Caption Channel Packet.

Figure 10 shows an example of a Caption Channel Packet containing two Standard Service Blocks and one Extended Service Block. The Caption Channel Packet size is 20 bytes and the Packet Sequence Number is 2.

The double lines indicate the start of byte pairs. The heavy double line indicates that the byte pair is marked “DTVCC Channel Packet Start”.

A Null Service Block Header (Figure 9) shall be inserted as the last Service Block in the Caption Channel Packet if space permits.

Packet Encoding equipment should null fill (i.e., zero out) the Caption Channel Packet buffer before any Service Blocks, which insures that a Null header is always present in Caption Channel Packets that are not completely filled.

Syntax:	msb	lsb	Packet Byte:
Packet Size: 0x0C, Seq# 2	1 0 0 0 1 0 1 0		0 (CCP Header)
Svc #: 1, BlkSz: 3	0 0 1 0 0 0 1 1		1 (Service 1)
			2
			3
			4
Svc #: 6, BlkSz: 4	1 1 0 0 0 1 0 0		5 (Service 6)
			6
			7
			8
			9
Ext Svc, BlkSz: 8 Svc#: 0x15 (21)	1 1 1 0 1 0 0 0		10 (Ext. Service 21)
	0 0 0 1 0 1 0 1		11
			12
			13
			14
			15
			16
			17
			18
			19

Figure 10 Service Blocks in a Caption Channel Packets (Example)

Note that it is allowable for more than one service block for a given service number to be present within a single CCP. In the above example, another Service #1 block could appear just following the Service #6 block or elsewhere in the packet.

6.3 Transport Constraints on Encapsulating Caption Data

The Caption Channel Packet and Service Block layers provide two structures for encapsulating caption data. Service Block boundaries are asynchronous with respect to the syntactic elements defined in Section 7. One service may be carried over several service blocks within a Caption Channel Packet, and, therefore, a syntactic element started in one service block may be finished in a subsequent block.

The Service Block and the Caption Channel Packet are both asynchronous with relation to the **cc_data()** structure.

Unlike the Service Block, the DTVCC Caption Channel Packet, described in Section 5.1, is a framing syntax, containing Caption Channel data. The start of a DTVCC Caption Channel Packet is a point where decoding of Caption Channel data may be resumed.

Every syntactic element shall be entirely contained in a single DTVCC Caption Channel Packet.

Aligning syntactic elements to Service Blocks as well as DTVCC Caption Channel Packets may simplify stream creation, and caption service encoders may do this by design; however, decoder designers should not assume that all streams have this additional alignment.

NOTE: Some decoders may reset if syntactic elements are left incomplete at the end of a DTVCC Caption Channel Packet, but there is no requirement for decoders to reset. If incomplete